ensia The National School of Artificial Intelligence المدرسة الوطنية العليا للذكاء الاصطناعي

2nd Year /Semester 3 (2024-2025)

Data Structures & Algorithms 2
Lab 6

Priority Queues

Objectives

- Efficient implementation of the priority queue ADT.
- Uses of priority queues.

Exercise 1

Write a program to take N elements and do the following:

- Insert the elements into a heap one by one.
- Build a heap in linear time.

Compare the running time of both algorithms for sorted, reverse-ordered, and random input.

Exercise 2

- Write a program to find all nodes less than some value X in a binary heap. Your algorithm should run in O(K), where K is the number of nodes output.

- Give an algorithm that finds an arbitrary item X in a binary heap using at most roughly 3N/4 comparisons.

Exercise 3

A min-max heap is a data structure that supports both deleteMin and deleteMax in O(log N). The structure is identical to a binary heap, but the heap order property ensure the following:

> ”First, for any node X at even depth, the element stored at X is smaller than the parent
> but larger than the grandparent (where this makes sense). Second, for any node X at odd depth, the element stored at X is larger than the parent but smaller than the grandparent (see Figure 1)”.

- How do we find the minimum and maximum elements?
- Give an algorithm to insert a new node into the min-max heap.
- Give an algorithm to perform deleteMin and deleteMax.
- Can you build a min-max heap in linear time?
- Let’s suppose we would like to support the operations deleteMin, deleteMax, and merge.

Propose a data structure to support all operations in O(log N) time.

Figure 1: Min-max heap